CAT
(Central Application Tracking)

美团点评基础架构中心 尤勇

自我介绍

- 尤勇
- 2010年加入美团点评 基础架构组
- 目前主要负责监控、移动接入层、slb等项目

大纲

- **CAT介绍**
- CAT设计
- 最佳实践

CAT介绍

- CAT(Central Application Tracking)是基于Java开发的实时监控平台，主要包括移动端监控、应用侧监控、核心网络层监控等。

- CAT是一个给提供实时监控告警，应用性能分析诊断的工具。

实时系统

- 1、客户端日志不落地
- 2、服务端流处理

- 整个系统从客户端产生消息到服务端产生实时报表延迟在**毫秒级别**

CAT的Logview

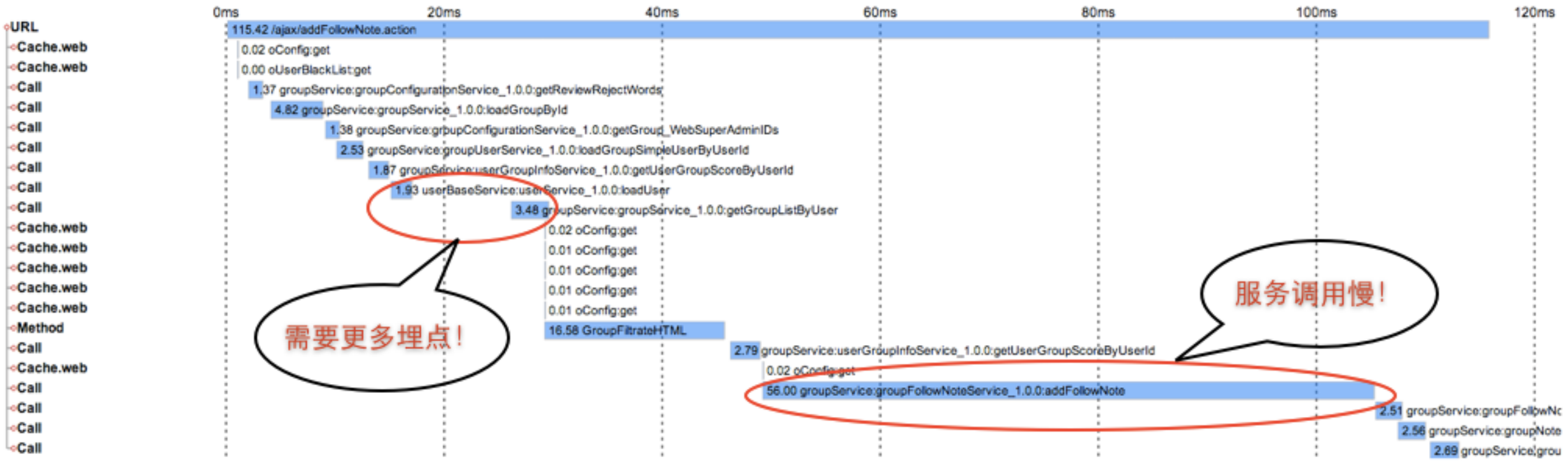
- 消息头
  - 版本号，消息ID，所属业务，IP，所在线程，根消息ID

```
t: Transaction Start
E: Event
T: Transaction End
A: Atomic Transaction
```

```
Transaction: 可嵌套
Event: 不可嵌套
Heartbeat: 不可嵌套
```

Type & timestamp	1st Category	2nd Category	Status	Duration & Attributes
t14:38:56.595	URL	t		
E14:38:56.595	URL.Server	cat.dianpingoa.com		RemoteIP= &Referer=http://cat.dianping
E14:38:56.595	URL.Method	HTTP/GET		/cat/r/t?domain=&date=2012101314&reportType=
A14:38:56.595	MVC	InboundPhase		0.06ms
A14:38:56.595	MVC	TransitionPhase		0.00ms
t14:38:56.595	MVC	OutboundPhase		
t14:38:56.595	ModelService	CompositeTransactionService		
A14:38:56.596	ModelService	RemoteTransactionService		1.06ms http:// :8080/cat/r/model/transact
A14:38:56.596	ModelService	RemoteTransactionService		0.86ms http:// :8080/cat/r/model/transac
A14:38:56.596	ModelService	RemoteTransactionService		1.89ms http:// :8080/cat/r/model/transac
A14:38:56.596	ModelService	RemoteTransactionService		1.79ms http:// :8080/cat/r/model/transac
A14:38:56.596	ModelService	RemoteTransactionService		27ms http:// :8080/cat/r/model/transacti
T14:38:56.622	ModelService	CompositeTransactionService		27ms request=ModelRequest[domain=Cat, period
T14:38:56.628	MVC	OutboundPhase		33ms
T14:38:56.628	URL	t		33ms module=r&in=t&out=t

可视化Logview


0ms
20ms
40ms
60ms
80ms
100ms
120ms
URL
Cache.web
Cache.web
Call
Call
Call
Call
Call
Cache.web
Cache.web
Cache.web
Cache.web
Cache.web
Method
Call
Cache.web
Call
Call
Call
115.42 /ajax/addFollowNote.action
0.02 oConfig:get
0.00 oUserBlackList:get
1.37 groupService:groupConfigurationService_1.0.0:getReviewRejectWords
4.82 groupService:groupService_1.0.0:loadGroupById
1.38 groupService:groupConfigurationService_1.0.0:getGroup_WebSuperAdminIDs
2.53 groupService:groupUserService_1.0.0:loadGroupSimpleUserByUserId
1.87 groupService:userGroupInfoService_1.0.0:getUserGroupScoreByUserId
1.93 userBaseService:userService_1.0.0:loadUser
3.48 groupService:groupService_1.0.0:getGroupListByUser
0.02 oConfig:get
0.01 oConfig:get
0.01 oConfig:get
0.01 oConfig:get
0.01 oConfig:get
16.58 GroupFiltrateHTML
2.79 groupService:userGroupInfoService_1.0.0:getUserGroupScoreByUserId
0.02 oConfig:get
56.00 groupService:groupFollowNoteService_1.0.0:addFollowNote
2.51 groupService:groupFollowNc
2.56 groupService:groupNote
2.69 groupService:grou
需要更多埋点!
服务调用慢!

分布式Logview

t15:00:44.023	URL	/ajax/addVote.action	
E15:00:44.023	URL	ClientInfo	RemoteIP=180.175.162.12
E15:00:44.023	URL	Payload	HTTP/POST /ajax/addVot
A15:00:44.023	Cache.web	oConfig:get	0.02ms finalKey=oConfig.(
A15:00:44.023	Cache.web	oUserBlackList:get	0.00ms finalKey=oUserBla
t15:00:44.026	Call	groupService:groupSurveyService_1.0.0:addVote	

[:: hide ::]

t15:00:43.967	Service	groupService:groupSurveyService_1.0.0:addVote	
E15:00:43.967	PigeonRequest	Payload	
t15:00:43.967	SQL	GroupSurvey.loadSurvey	
E15:00:43.967	SQL.Method	Select	
E15:00:43.968	SQL.Database	jdbc:mysql:// [redacted]	?characterEncoding=(
T15:00:43.967	SQL	GroupSurvey.loadSurvey	
t15:00:43.968	Call	userBaseService:userService_1.0.0:loadUser	

[:: hide ::]

t15:00:44.089	Service	userBaseService:us	
E15:00:44.089	PigeonRequest	Payload	
A15:00:44.089	Cache.memcached	eUserAtUC:get	
T15:00:44.089	Service	userBaseService:us	

[:: show ::]

T15:00:43.970	Call	userBaseService:userService_1.0.0:loadUser	
A15:00:43.970	Cache.memcached	oUserGroupScore:get	
t15:00:43.975	SQL	GroupSurvey.addVote	
E15:00:43.975	SQL.Method	Execute	
E15:00:44.244	SQL.Database	jdbc:mysql:// [redacted]	?characterEncoding=(
T15:00:44.243	SQL	GroupSurvey.addVote	
T15:00:44.244	Service	groupService:groupSurveyService_1.0.0:addVote	

[:: show ::]

T15:00:44.305	Call	groupService:groupSurveyService_1.0.0:addVote	279ms CallType=sync
T15:00:44.307	URL	/ajax/addVote.action	284ms

应用监控报表（APM）

报表	说明
Transaction	一段代码运行时间、次数
Event	一行代码的执行次数
Problem	系统可能出现的异常，包括访问较慢的程序等
Hearbeat	JVM内部一些状态信息，Memory，Thread等
Matrix	一个请求调用链路统计
Cross	SOA系统用关于RPC调用的报表
Cache	缓存使用分析统计
Dependency	系统之间实时调用数据信息，远程服务、数据库、缓存等
…	…

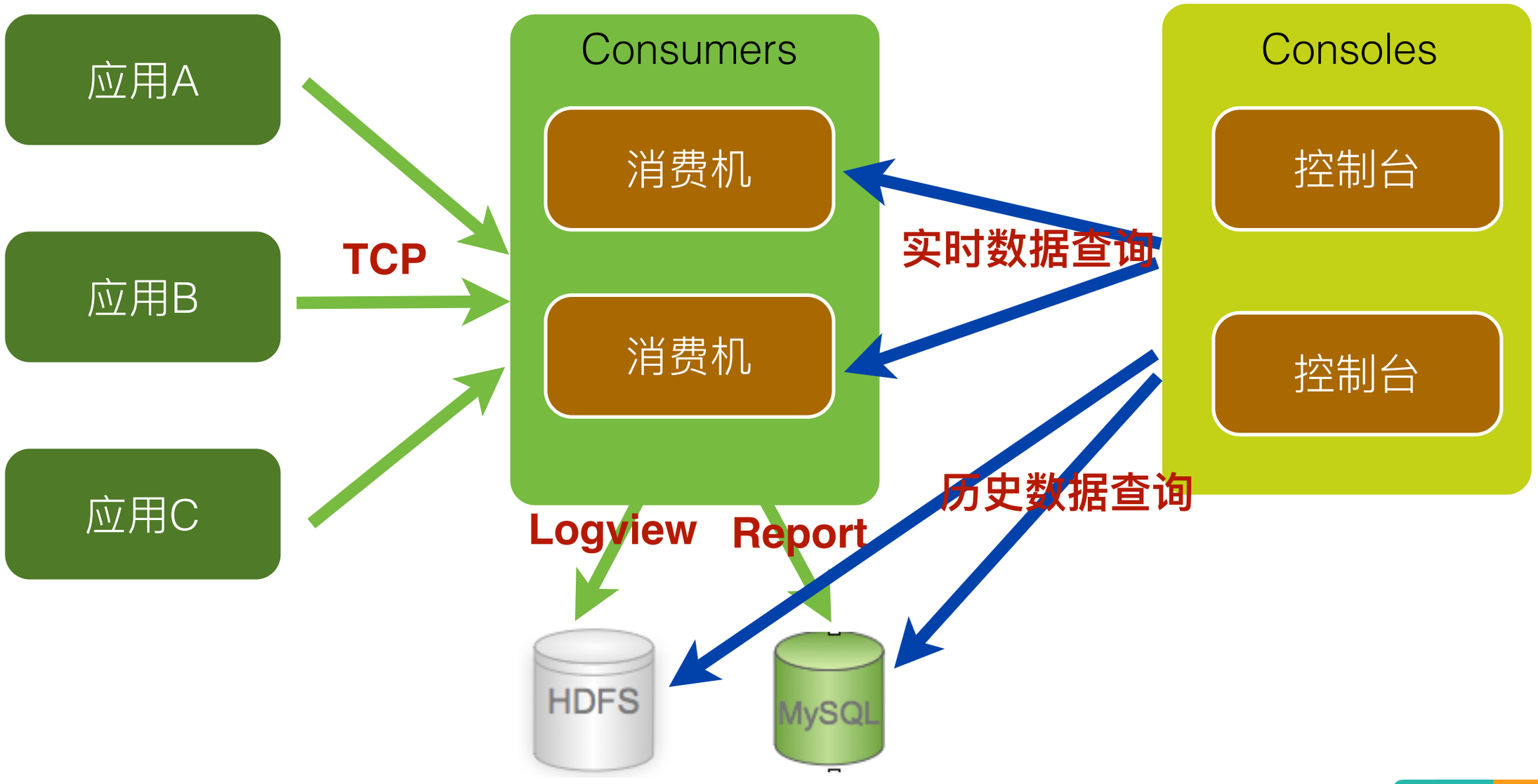
大纲

- **CAT历程**
- **CAT设计**
- **最佳实践**

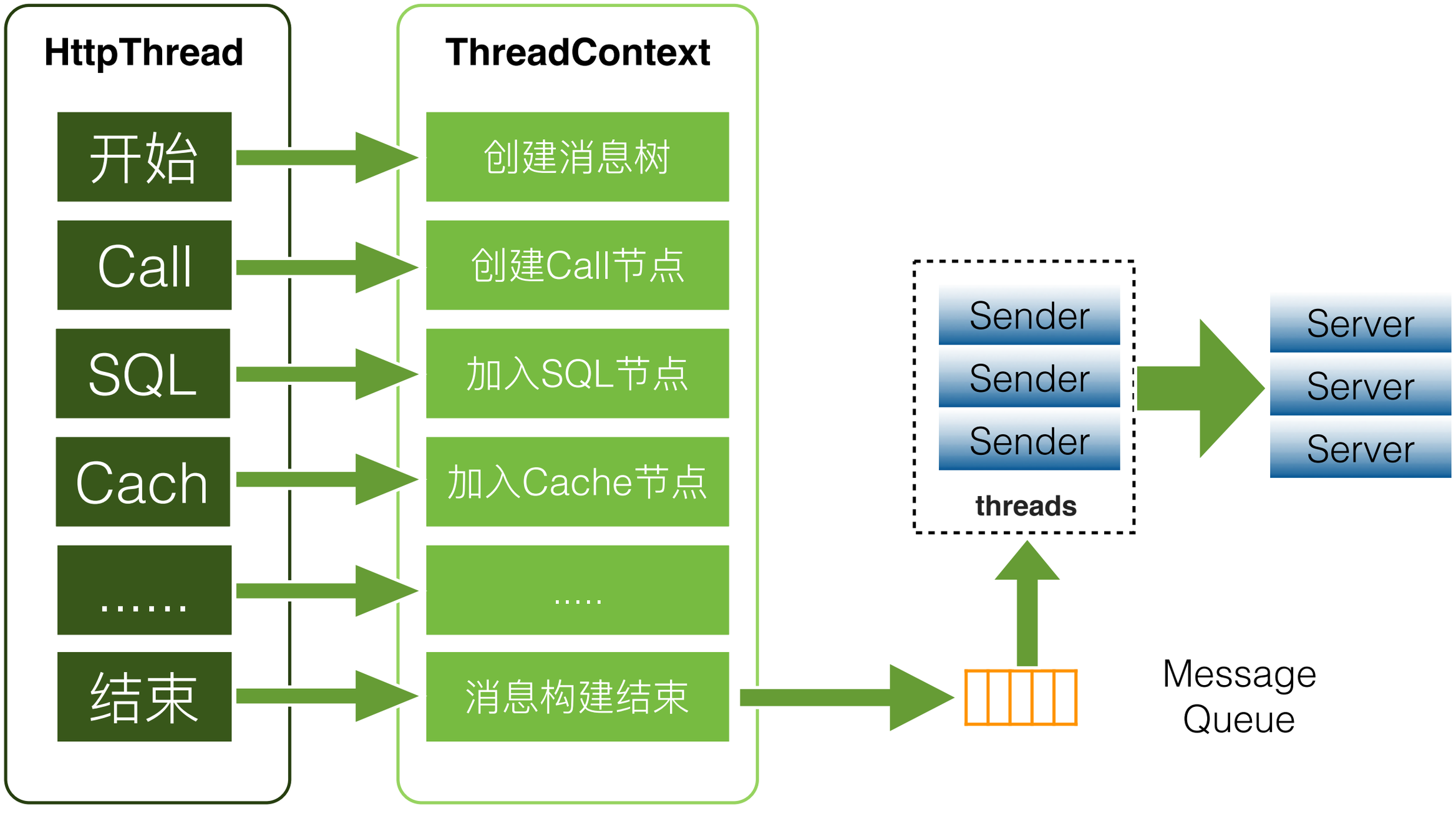
CAT设计

- 整体设计
- 客户端设计
- 服务端设计

整体设计


应用A
应用B
应用C
TCP
Consumers
消费机
消费机
Consoles
控制台
控制台
实时数据查询
历史数据查询
Logview
Report
HDFS
MySQL

客户端设计


HttpThread
开始
Call
SQL
Cach
.....
结束
ThreadContext
创建消息树
创建Call节点
加入SQL节点
加入Cache节点
.....
消息构建结束
Sender
Sender
Sender
threads
Server
Server
Server
Message
Queue

客户端重点

- 内存开销
  - 由于埋点问题，消息足够大
- CPU开销
  - 构建消息足够轻量，开销减低在2%
- 客户端没有做压缩
- 基于netty实现消息传输

遇到问题-IO

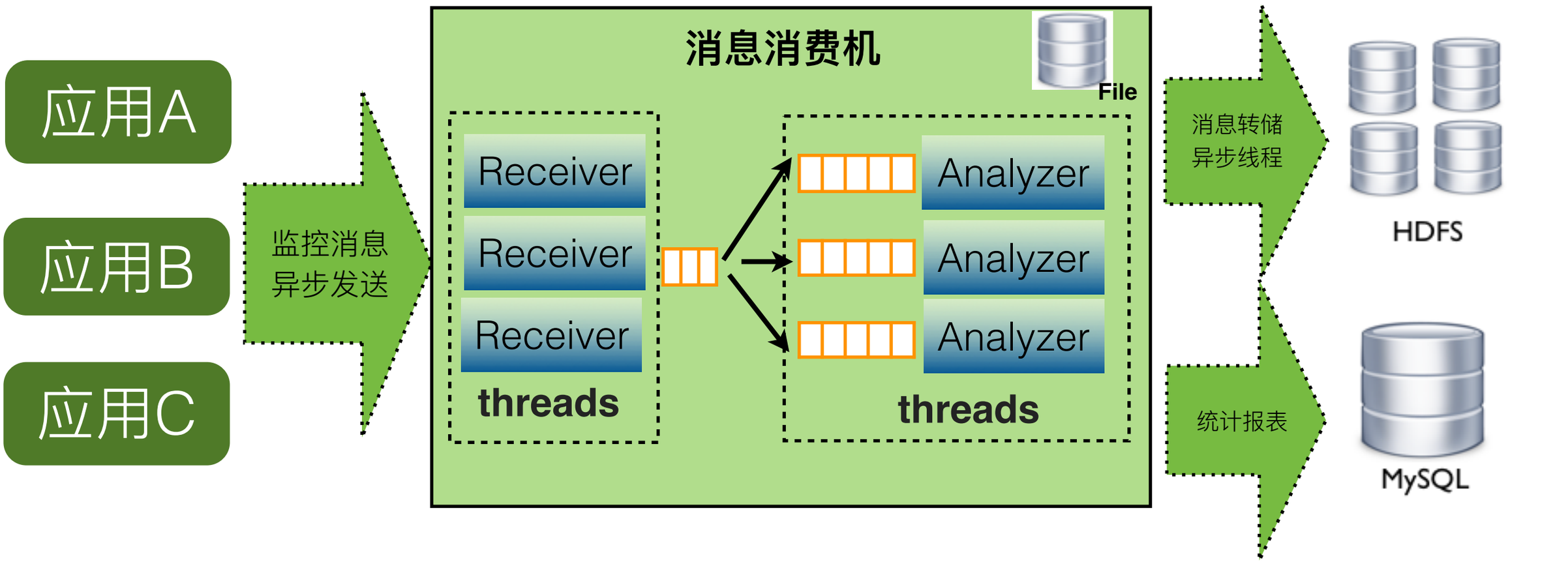
- java message tree id的生成，java MappedByteBuffer需要做持久化
- 业务主线程的使用
- 在任何时候客户端都是需要考虑极端情况cpu或者io的开销

遇到问题-Memory

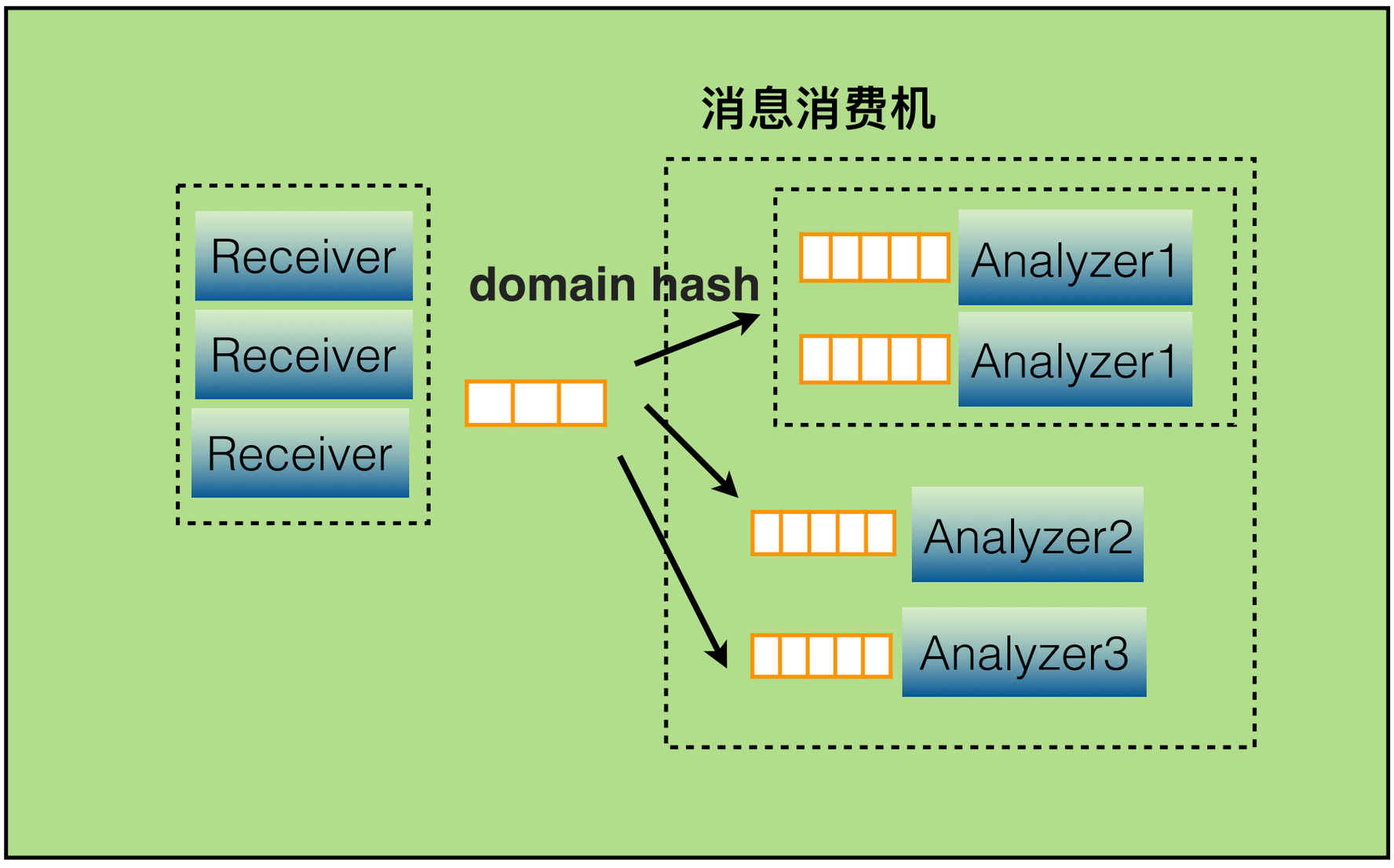
- MessageTree的内存占用太大，极端情况下，一个messageTree里面上万个节点

- 在任何时候客户端都是需要考虑极端情况内存的开销

服务端设计


应用A
应用B
应用C
监控消息
异步发送
消息消费机
File
Receiver
Receiver
Receiver
threads
Analyzer
Analyzer
Analyzer
threads
消息转储
异步线程
HDFS
统计报表
MySQL

当Analyzer处理来不及



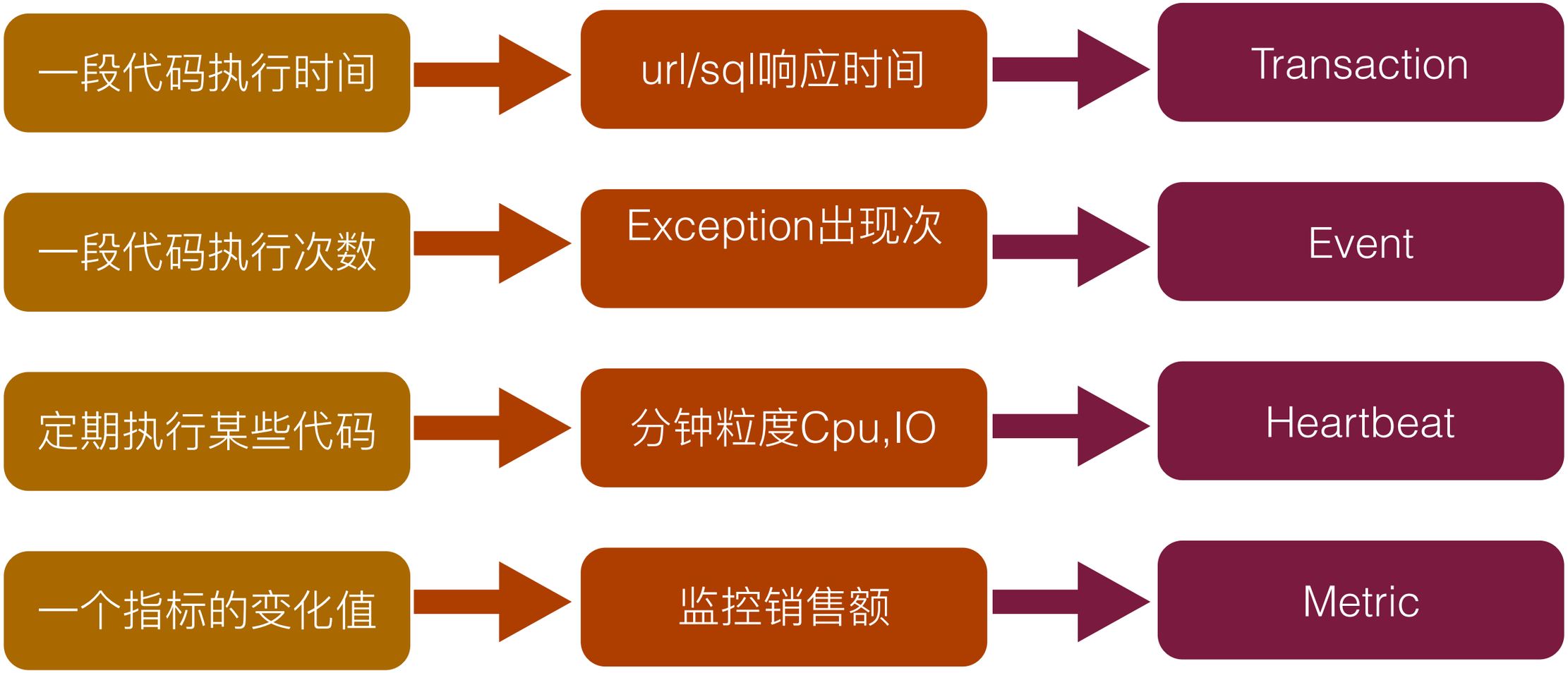
服务端重点

- 监控建模
- 报表建模
- CPU优化
- 负载均衡
- 数据存储
- 内存以及系统问题

建模

- 监控领域数据模型
- 数据报表模型

监控建模


一段代码执行时间
url/sql响应时间
Transaction
一段代码执行次数
Exception出现次
Event
定期执行某些代码
分钟粒度Cpu,IO
Heartbeat
一个指标的变化值
监控销售额
Metric

KeyValue的方式

- 后续扩展性较好
- 后续配置成本很高
- 后续计算成本很高

报表

- Transaction
- Event
- Problem
- Heartbeat
- ……

报表建模

- 目标模型定义
- 访问、转换和合并
- 模型持久化
- XML, JSON, Binary...
- 代码生成

```xml
<?xml version="1.0" encoding="UTF-8"?>
<model>
      <entity name="transaction-report" root="true">
            <attribute name="domain" value-type="String" key="true" />
            <attribute name="startTime" value-type="Date" />
            <attribute name="endTime" value-type="Date" />
            <entity-ref name="machine" type="map" names="machines" />
      </entity>
      <entity name="machine">
            <attribute name="ip" value-type="String" key="true"/>
            <entity-ref name="type" type="map" names="types" />
      </entity>
      <entity name="type">
            <attribute name="id" value-type="String" key="true" />
            <attribute name="total-count" value-type="int" />
            <attribute name="fail-count" value-type="int" />
            <attribute name="min" value-type="double" />
            <attribute name="max" value-type="double" />
            <attribute name="sum" value-type="double" />
            <attribute name="sum2" value-type="double" />
            <element name="success-message" value-type="String" />
            <element name="fail-message" value-type="String" />
            <entity-ref name="name" type="map" names="names" />
      </entity>
      ...
</model>
```

```java
public interface IVisitor {
   public void visitTransactionReport(TransactionReport transactionReport);
   public void visitMachine(Machine machine);
   public void visitType(TransactionType type);
   public void visitName(TransactionName name);
   public void visitRange(Range range);
   public void visitDuration(Duration duration);
}
```

模型遍历

```
public abstract class BaseVisitor implements IVisitor {
    @Override
    public void visitAllDuration(AllDuration allDuration) {
    }

    @Override
    public void visitDuration(Duration duration) {
    }

    @Override
    public void visitMachine(Machine machine) {
        for (TransactionType type : machine.getTypes().values()) {
            visitType(type);
        }
    }

    @Override
    public void visitName(TransactionName name) {
        for (Range range : name.getRanges().values()) {
            visitRange(range);
        }

        for (Duration duration : name.getDurations().values()) {
            visitDuration(duration);
        }

        for (AllDuration allDuration : name.getAllDurations().values()) {
            visitAllDuration(allDuration);
        }
    }

    @Override
    public void visitRange(Range range) {
    }
```

模型合并

```
public class TransactionReportMerger extends DefaultMerger {
   public TransactionReportMerger(TransactionReport transactionReport) {
      super(transactionReport);
   }

   @Override
   public void mergeDuration(Duration old, Duration duration) {
      old.setCount(old.getCount() + duration.getCount());
      old.setValue(duration.getValue());
   }

   @Override
   public void mergeMachine(Machine old, Machine machine) {
   }

   @Override
   public void mergeName(TransactionName old, TransactionName other) {
      long totalCountSum = old.getTotalCount() + other.getTotalCount();
      if (totalCountSum > 0) {
         double line95Values = old.getLine95Value() * old.getTotalCount() + other.getLine95Value()
               * other.getTotalCount();
         double line99Values = old.getLine99Value() * old.getTotalCount() + other.getLine99Value()
               * other.getTotalCount();

         old.setLine95Value(line95Values / totalCountSum);
         old.setLine99Value(line99Values / totalCountSum);
      }

      old.setTotalCount(totalCountSum);
      old.setFailCount(old.getFailCount() + other.getFailCount());
      old.setTps(old.getTps() + other.getTps());

      if (other.getMin() < old.getMin()) {
         old.setMin(other.getMin());
      }

      if (other.getMax() > old.getMax()) {
         old.setMax(other.getMax());
      }
```

cpu优化

```
protected static class DateHelper {
   private BlockingQueue<SimpleDateFormat> m_formats = new ArrayBlockingQueue<SimpleDateFormat>(20);

   private Map<String, Long> m_map = new ConcurrentHashMap<String, Long>();

   public String format(long timestamp) {
      SimpleDateFormat format = m_formats.poll();

      if (format == null) {
         format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss.SSS");
         format.setTimeZone(TimeZone.getTimeZone("GMT+8"));
      }

      try {
         return format.format(new Date(timestamp));
      } finally {
         if (m_formats.remainingCapacity() > 0) {
            m_formats.offer(format);
         }
      }
   }
```

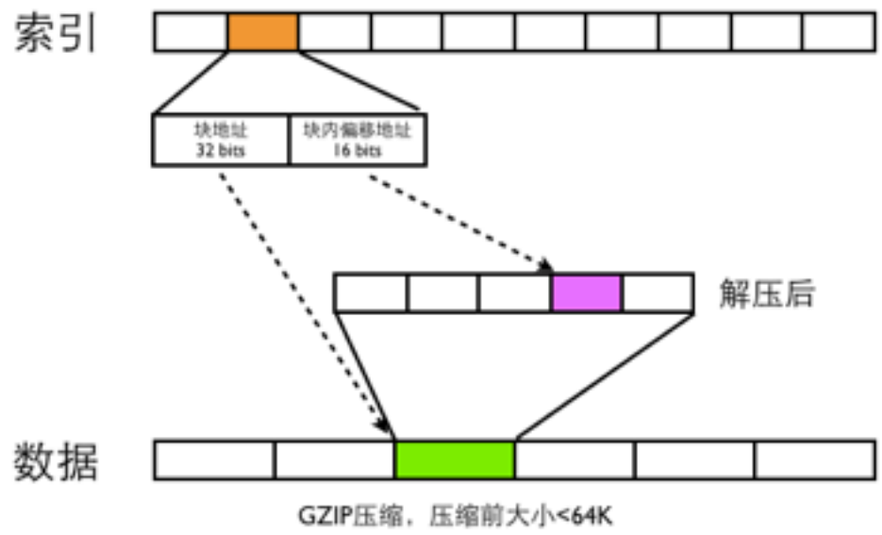
```
public long parse(String str) {
   int len = str.length();
   String date = str.substring(0, 10);
   Long baseline = m_map.get(date);

   if (baseline == null) {
      try {
         SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");

         format.setTimeZone(TimeZone.getTimeZone("GMT+8"));
         baseline = format.parse(date).getTime();
         m_map.put(date, baseline);
      } catch (ParseException e) {
         return -1;
      }
   }

   long time = baseline.longValue();
   long metric = 1;
   boolean millisecond = true;

   for (int i = len - 1; i > 10; i--) {
      char ch = str.charAt(i);

      if (ch >= '0' && ch <= '9') {
         time += (ch - '0') * metric;
         metric *= 10;
      } else if (millisecond) {
         millisecond = false;
      } else {
         metric = metric / 100 * 60;
      }
   }
   return time;
}
```

数据存储

- 顺序写、随机读
- 批量压缩提高压缩率

数据存储

- 消息ID：**ShopWeb-0a010680-375030-2**
  - 消息可能的存储路径
    - /2012/10/13/14/ShopSerivce-ShopWeb-10.1.6.1
    - /2012/10/13/14/ShopService-ShopWeb-10.1.6.2
    - 375030 => 2012-10-13 14:00:00
    - ShopService => 消息被记录的domain
    - 10.1.6.1/2 => 消息被处理的机器IP
    - 0a010680 => 10.1.6.128 用于保证消息ID唯一性



内存困惑

- 1、transaction report，event report里面的name无限个节点
- 2、swap off
- 3、numactl --interleave=all
- 4、非jvm线程的影响

大纲

- **CAT介绍**
- **CAT设计**
- **最佳实践**

CAT历程

- 2011-11月份　启动
- 2012-3月份　MVP模型
- 2012-6月份　正式上线
- 2012-12月份　150+应用　500+服务器
- 2013-12月份　400+应用　1500+服务器
- 2014-12月份　800+应用　3000+服务器
- 2015-9月份　1500+应用 7000+服务器
- 2016-6月份　2600+应用 12000+服务器

MVP版本

- Demo 1个月
- MVP 3个月
- 重点解决最急迫的一个问题

小白鼠客户

- 典型客户
- vip服务

一些不和谐的声音

- 客户端
- 业务的挑战（可靠，性能）
- 领导的挑战（当***时候，加一个动态开关）

上线以及后续

- 独立快速发布（项目初期）
- 灰度发布（项目中后期）
- 问题排查（mat）

不仅仅是code

- 不同角色如何使用系统
- 系统如何运维
- 系统如何推广

数据质量

- 数据质量
- sql框架、缓存框架、rpc框架、web框架
- 数据质量决定了监控质量

单机开发环境

- jetty server
- hdfs依赖
- mysql依赖

```java
@RunWith(JUnit4.class)
public class TestServer extends JettyServer {
    public static void main(String[] args) throws Exception {
        TestServer server = new TestServer();
        System.setProperty("devMode", "true");
        server.startServer();
        server.startWebApp();
        server.stopServer();
    }

    @Before
    public void before() throws Exception {
        System.setProperty("devMode", "true");
        super.startServer();
    }

    @Override
    protected String getContextPath() {
        return "/cat";
    }

    @Override
    protected int getServerPort() {
        return 2281;
    }

    @Override
    protected void postConfigure(WebAppContext context) {
        context.addFilter(GzipFilter.class, "/*", Handler.ALL);
    }

    @Test
    public void startWebApp() throws Exception {
        // open the page in the default browser
        display("/cat/r");
        waitForAnyKey();
    }
}
```

最难的事情

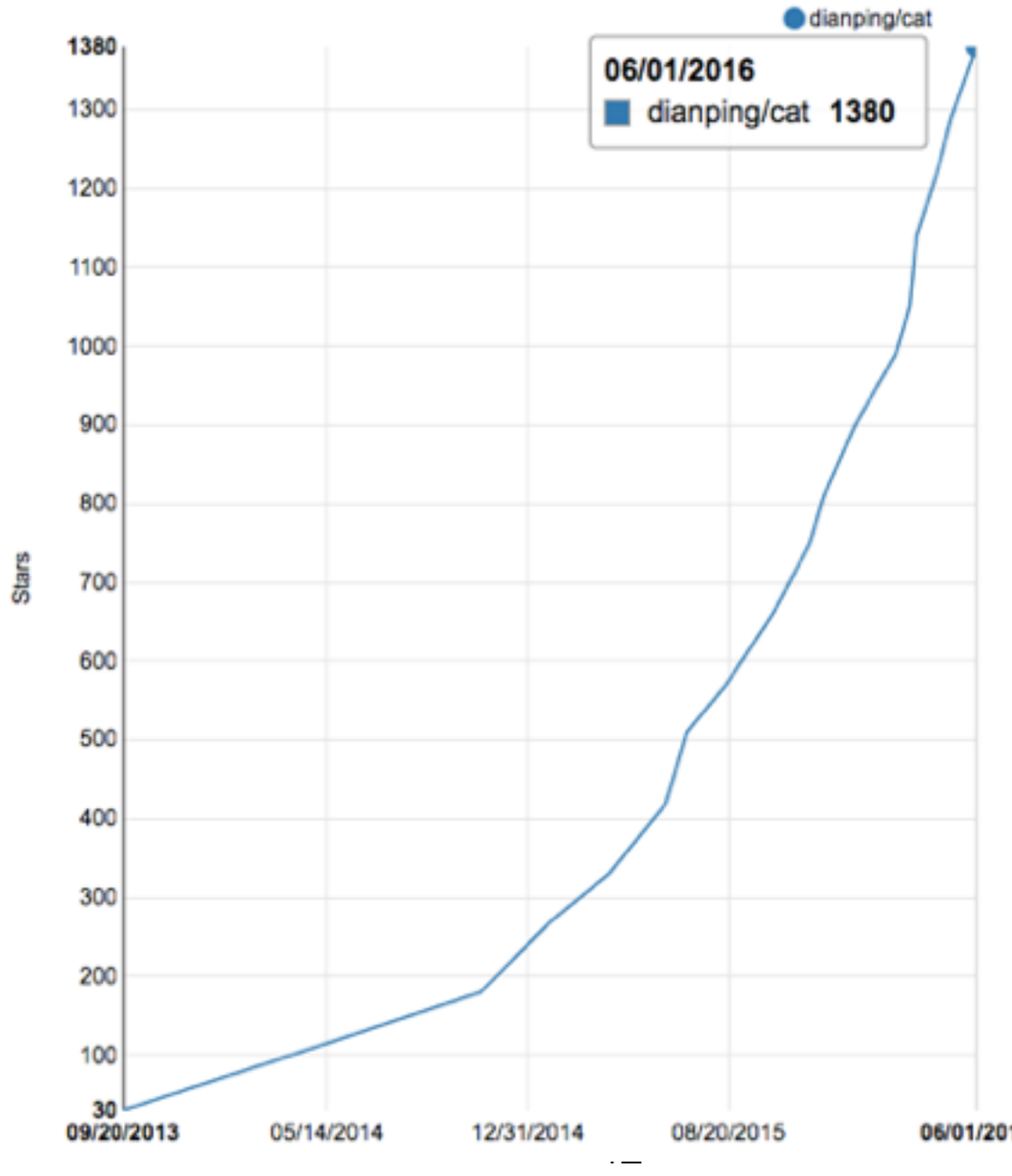
- 项目上线推动
  - 如何推动整个项目上线（2-3人）
  - 部门之间沟通问题
  - 后续的支持和培训

其他需求

- 监控的scope
- 其他的需求
- 系统开放生态

Open Source

- http://github.com/dianping/cat



项目合作

- https://github.com/dianping/cat/issues/753

QA

- Thanks